

*Proceedings of the 27th International Conference on Probabilistic, Combinatorial  
and Asymptotic Methods for the Analysis of Algorithms  
Kraków, Poland, 4-8 July 2016*

# *Robin Hood Hashing really has constant average search cost and variance in full tables*

Patricio V. Poblete<sup>1†</sup>, Alfredo Viola<sup>2‡</sup>

<sup>1</sup>*Dept. of Computer Science, University of Chile, Chile*

<sup>2</sup>*Universidad de la República, Uruguay*

---

Thirty years ago, the Robin Hood collision resolution strategy was introduced for open addressing hash tables, and a recurrence equation was found for the distribution of its search cost. Although this recurrence could not be solved analytically, it allowed for numerical computations that, remarkably, suggested that the variance of the search cost approached a value of 1.883 when the table was full. Furthermore, by using a non-standard mean-centered search algorithm, this would imply that searches could be performed in expected constant time even in a full table.

In spite of the time elapsed since these observations were made, no progress has been made in proving them. In this paper we introduce a technique to work around the intractability of the recurrence equation by solving instead an associated differential equation. While this does not provide an exact solution, it is sufficiently powerful to prove a bound for the variance, and thus obtain a proof that the variance of Robin Hood is bounded by a small constant for load factors arbitrarily close to 1. As a corollary, this proves that the mean-centered search algorithm runs in expected constant time.

We also use this technique to study the performance of Robin Hood hash tables under a long sequence of insertions and deletions, where deletions are implemented by marking elements as *deleted*. We prove that, in this case, the variance is bounded by  $1/(1 - \alpha) + O(1)$ , where  $\alpha$  is the load factor.

To model the behavior of these hash tables, we use a unified approach that can be applied also to study the First-Come-First-Served and Last-Come-First-Served collision resolution disciplines, both with and without deletions.

**Keywords:** Robin Hood Hashing, full tables, constant variance, constant expected search time

---

## 1 Introduction

In 1986, Celis *et al* [3, 4] introduced the Robin Hood collision resolution strategy for open addressing hash tables. Under this discipline, collisions are decided in favor of the element that is farthest from its home location. While this does not change the expected search cost, it turns out to have a dramatic effect

---

<sup>†</sup>Supported in part by NIC Chile

<sup>‡</sup>This work has been partially supported by Project CSIC I+D "Combinatoria Analítica y aplicaciones en criptografía, comunicaciones y recuperación de la información", fondos 2015-2016.

on its *variance*. In effect, unlike other disciplines where the variance tends to infinity as the table becomes full, the variance of Robin Hood seems to remain constant, and very small. This fact, conjectured from numerical computations, has not been proved in the years since it was observed, and is the main focus of our work. This problem has been hard to solve because the distribution of the search cost obeys a nonlinear recurrence equation for which no successful line of attack has been found.

To show the kind of recurrence involved, we quote now Theorem 3.1 from [3] (our notation will be slightly different):

**Theorem 3.1** *In the asymptotic model for an infinite Robin Hood hash table with load factor  $\alpha$  ( $\alpha < 1$ ), the probability  $p_i(\alpha)$  that a record is placed in the  $i$ -th or further position in its probe sequence is equal to*

$$p_1(\alpha) = 1, \quad p_{i+1}(\alpha) = 1 - \left( \frac{1-\alpha}{\alpha} \right) \left( e^{\alpha(p_1(\alpha) + \dots + p_i(\alpha))} \right). \quad (1)$$

They then go on to define another function  $r_i(\alpha) = \alpha(p_i(\alpha) + \dots + p_\infty(\alpha))$ , in terms of which the variance can be expressed as

$$V(\alpha) = \frac{2}{\alpha} \sum_{i=1}^{\infty} r_i(\alpha) + \frac{\ln(1-\alpha)}{\alpha} - \frac{\ln^2(1-\alpha)}{\alpha^2}. \quad (2)$$

They show that  $r_i(\alpha)$  satisfies the following recurrence equation:

$$r_i(\alpha) - r_{i+1}(\alpha) = 1 - e^{-r_i(\alpha)} \quad (3)$$

with  $r_1(\alpha) = -\ln(1-\alpha)$ . By leaving the “ $(\alpha)$ ” implicit and using the  $\Delta$  operator (defined as  $\Delta r_i = r_{i+1} - r_i$ ), this can be rewritten as  $\Delta r_i = f(r_i)$  where  $f$  is the function  $f(x) = -1 + e^{-x}$ .

This seemingly simpler equation has, nonetheless, so far remained unsolved.

In this paper, we will introduce a technique applicable to equations of this form, and we will use it first to prove a bound on the variance of Robin Hood hashing. Then we will use it to study another recurrence equation of the same type arising from the problem of hashing with deletions.

## 2 Modeling hashing algorithms

In this paper we will study the search cost of a random element in a hash table, using the *random probing model*. This is an open addressing hashing scheme in which collisions are resolved by additional probes into the table. The sequence of these probes are considered to be random and depends only on the value of the key. The difference with uniform probing is that positions may be repeated in this sequence. We use the *asymptotic model* for a hash table with load factor  $\alpha$  [9, 8, 4, 12], where we assume that the number of keys  $n$  and the table size  $m$  both tend to infinity, maintaining constant their ratio  $\alpha = n/m$ .

Each element has associated with it an infinite probe sequence consisting of i.i.d. integers uniformly distributed over  $\{0, \dots, m-1\}$ , representing the consecutive places of probes for that element. The probe sequence for element  $x$  is denoted by  $h_1(x), h_2(x), h_3(x), \dots$ . Elements are inserted sequentially into the table. If element  $x$  is placed in position  $h_j(x)$ , then we say that element  $x$  has age  $j$ , as it requires  $j$  probes to reach the element in case of a search. When an element  $x$  of age  $j$  and an element  $y$  of age  $k$  compete for the same slot ( $h_j(x) = h_k(y)$ ), a collision resolution strategy is needed.

In the standard method, a collision is resolved in favor of the incumbent key, so the incoming key continues probing to its next location. We call this a First-Come-First-Served (FCFS) collision resolution

discipline. Several authors [2, 1, 7] observed that a collision could be resolved in favor of *any* of the keys involved, and used this additional degree of freedom to decrease the expected search time in the table.

Celis *et al* [3, 4] were the first to observe that collisions could be resolved having instead *variance reduction* as a goal. They defined the Robin Hood (RH) heuristic, in which each collision occurring during an insertion is resolved in favor of the key that is farthest away from its home location (i.e., oldest in terms of *age*). Later, Poblete and Munro [14] defined the Last-Come-First-Served heuristic, where collisions are resolved in favor of the *incoming* key.

In both cases, the variance is reduced, and this can be used to speed up searches by replacing the standard search algorithm by a *mean-centered* one that first searches in the vicinity of where we would expect the element to have *drifted* to, rather than in its initial probe location. This *mean-centered* approach was introduced in [3] (and called “organ-pipe search”) to speed up successful searches in the Robin Hood heuristic, with expected cost bounded by the standard deviation of this random variable. Numerical computations in [3] suggest that for full tables the variance of the search cost for RH is constant, but no formal proof is given.

In this paper we formally settle this conjecture, by proving that this is in fact the case, and give an explicit upper bound (although not as tight as the numerical results seem to suggest). As a consequence we prove that the mean-centered searching algorithm in [3] has constant expected cost for full tables.

In section 4 we extend this approach to perform the analysis of hashing with deletions. Deletions in open addressing hash tables are often handled by marking the cells as *deleted* instead of *empty*, because otherwise the search algorithm might fail to find some of the keys. The space used by deleted cells may be reused by subsequent insertions. Intuitively, search times should deteriorate as tables become contaminated with deleted cells and, as Knuth[11] points out, in the long run the average successful search time should approach the average *unsuccessful* search time.

In this paper we analyze the effect of a long sequence of insertions and deletions in the asymptotic regime ( $\alpha$ -full tables with  $0 \leq \alpha < 1$ ) and prove a bound for the variance of RH with deletions that is close to numerical results.

There is an alternative algorithm designed to keep variance low in the presence of deletions. This method marks cells as deleted, but keeps the key values (these cells are called *tombstones*). In this paper we do not study the algorithm with tombstones. We note that [12] derives equations for this algorithm, but only obtains numerical solutions.

### 3 Analysis without deletions

To analyze the cost of searching for a random element, we begin by presenting a general framework, based on the one used in [5]. This framework applies also to FCFS and LCFS, but in this paper we use it to analyze RH, which has been a long standing open problem. As stated before, we use the asymptotic model for a hash table with load factor  $\alpha$  and random probing.

Under this model, if collisions are resolved without “looking ahead” in the table, the cost of inserting a random element is 1 plus a random variable that follows a geometric distribution with parameter  $1 - \alpha$ , and therefore its expected cost is  $1/(1 - \alpha)$ , independently of the collision resolution discipline used.

Let  $p_i(\alpha)$  be the probability that a randomly chosen key has age  $i$  when the table has load factor  $\alpha$ .

Suppose we insert a new element. Depending on the insertion discipline used, a number of keys will change locations and therefore increase their ages as a consequence of the arrival of the new element. Let us call  $t_i(\alpha)$  the expected number of probes made by keys of age  $i$  during the course of the insertion. It is

easy to see that

$$t_1(\alpha) = 1, \quad \sum_{i \geq 1} t_i(\alpha) = \frac{1}{1-\alpha}. \quad (4)$$

Before the insertion, the expected number of keys of age  $i$  is  $\alpha m p_i(\alpha)$ . After the insertion, it is

$$(\alpha m + 1)p_i(\alpha + \frac{1}{m}) = \alpha m p_i(\alpha) + t_i(\alpha) - t_{i+1}(\alpha) \quad (5)$$

If we write  $\Delta\alpha = 1/m$  and  $q_i(\alpha) = \alpha p_i(\alpha)$ , this equation becomes

$$\frac{q_i(\alpha + \Delta\alpha) - q_i(\alpha)}{\Delta\alpha} = t_i(\alpha) - t_{i+1}(\alpha) \quad (6)$$

and, as  $\Delta\alpha \rightarrow 0$  (i.e.  $m \rightarrow \infty$ ),

$$\partial_\alpha q_i(\alpha) = t_i(\alpha) - t_{i+1}(\alpha), \quad (7)$$

where  $\partial_\alpha$  denotes a derivative with respect to  $\alpha$ , and with the initial condition  $q_i(0) = 0$ .

We introduce a notation that we will use throughout the paper. For any sequence  $a_i$  we define its *tail*  $\bar{a}_i$  as

$$\bar{a}_i = \sum_{j \geq i} a_j. \quad (8)$$

Using this, equation (7) can be rewritten as

$$\partial_\alpha \bar{q}_i(\alpha) = t_i(\alpha). \quad (9)$$

We note that this equation is valid for all three collision resolution strategies, and it generalizes formula (10) in [12], where it is proved only for RH.

The mean of the search cost can be obtained using the tail notation, as

$$\mu_\alpha = \bar{\bar{p}}_1(\alpha) = \frac{1}{\alpha} \bar{\bar{q}}_1(\alpha) \quad (10)$$

and the variance as

$$\sigma_\alpha^2 = 2\bar{\bar{p}}_1(\alpha) - \mu_\alpha - \mu_\alpha^2 = \frac{2}{\alpha} \bar{\bar{q}}_1(\alpha) - \mu_\alpha - \mu_\alpha^2 \quad (11)$$

We note that we can already compute the expected search cost, without needing to know the exact form of the function  $t_i(\alpha)$ . Taking tails in both sides of (9), we have  $\partial_\alpha \bar{\bar{q}}_i(\alpha) = \bar{t}_i(\alpha)$ .

Now setting  $i = 1$  and using (10), we obtain  $\partial_\alpha (\alpha \mu_\alpha) = \frac{1}{1-\alpha}$ , and from this we obtain

$$\mu_\alpha = \frac{1}{\alpha} \ln \frac{1}{1-\alpha} \quad (12)$$

independently of the collision resolution discipline used.

The fact that the mean search cost is independent of the collision resolution discipline used does not necessarily carry over to higher moments or to the distribution of the search cost. To compute them, we need to know the  $t_i(\alpha)$  for the specific discipline.

For RH, a key will be forced to try its  $(i + 1)$ st probe location or higher each time there is a collision between an incoming key of age  $i$  or higher and another key in the table that is also of age  $i$  or higher. Therefore, and leaving the “ $(\alpha)$ ” implicit, to simplify notation, we have:

$$\bar{t}_{i+1} = \bar{t}_i \bar{q}_i \quad (13)$$

Together with equation (7) this implies  $\partial_\alpha \bar{q}_i = (1 - \bar{q}_i) \partial_\alpha \bar{q}_i$ . Then, after integrating both sides of the equation we have  $\ln \frac{1}{1 - \bar{q}_i} = \bar{q}_i$  from where we obtain  $\bar{q}_i = 1 - e^{-\bar{q}_i}$ . Moreover, by expressing  $\bar{q}$  as the difference of two  $\bar{q}$ , we arrive at

**Theorem 1** *Under the asymptotic model for an infinite hash table with random probing, and Robin Hood collision resolution discipline, the double tail of the probability distribution of the search cost of a random element satisfies the recurrence*

$$\bar{q}_{i+1} = \bar{q}_i - 1 + e^{-\bar{q}_i} \quad (14)$$

with the initial condition  $\bar{q}_1 = \ln \frac{1}{1 - \alpha}$ .  $\square$

This is exactly equation (3) that we quoted from [3], but we obtained it through a completely different derivation. As we mentioned before, numerical computations performed in [4] indicate that as  $\alpha \rightarrow 1$ , the variance converges to a small constant, approximately equal to 1.883.

### 3.1 Bounding the variance of RH

Since we are interested in the behavior of the method as  $\alpha \rightarrow 1$ , we will introduce a variable  $\beta$  defined as  $\beta = \frac{1}{1 - \alpha}$ , so that  $\alpha = 1 - \frac{1}{\beta} \rightarrow 1$  as  $\beta \rightarrow \infty$ . Now we rewrite equation (14) as

$$\Delta \bar{q}_i = -1 + e^{-\bar{q}_i}, \quad (15)$$

with  $\bar{q}_1 = \ln \beta$ . This equation is of the form

$$\Delta \bar{q}_i = f(\bar{q}_i), \quad (16)$$

where  $f$  is the function  $f(x) = -1 + e^{-x}$ . This recurrence equation seems very hard to solve exactly, but we will be able to obtain useful information about its solution by studying instead the differential equation

$$Q'(x) = f(Q(x)) \quad (17)$$

with the same initial condition  $Q(1) = \ln \beta$ . The solution to this equation is

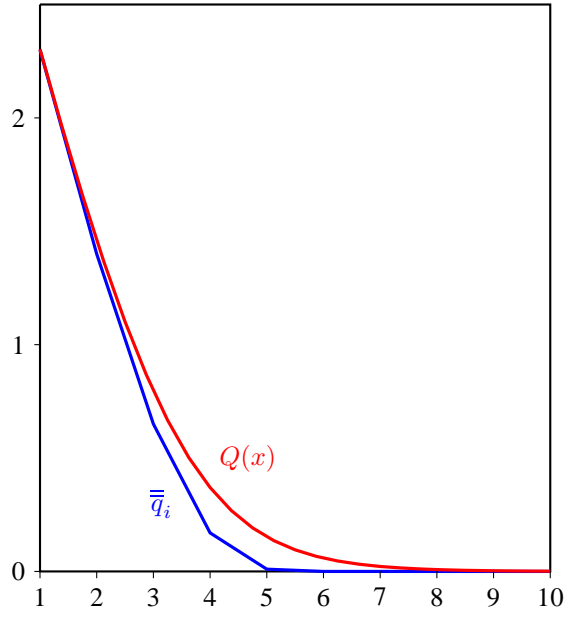
$$Q(x) = \ln(\beta - 1 + e^{x-1}) - x + 1. \quad (18)$$

Figure 1 compares the solution  $\bar{q}_i$  (polygonal line) of recurrence equation (16) to the solution  $Q(x)$  (smooth line) of differential equation (17). This plot suggests that  $Q(i)$  is an upper bound for  $\bar{q}_i$ . This is true, and will follow from the following lemma.

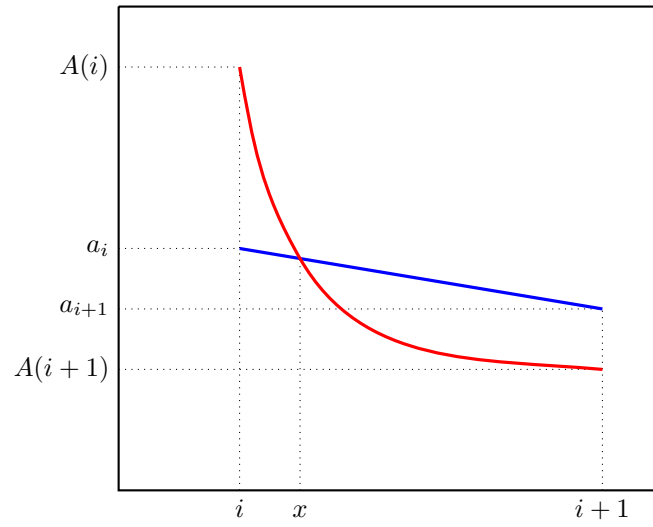
**Lemma 1** *Let  $a_i$  satisfy the recurrence equation  $\Delta a_i = f(a_i)$ , and  $A(x)$  satisfy the differential equation  $A'(x) = f(A(x))$ , where  $f : [0, +\infty) \rightarrow (-\infty, 0]$  is a decreasing function. Then*

$$A(i) \geq a_i \implies A(i + 1) \geq a_{i+1} \quad (19)$$

for all  $i \geq 1$ .



**Fig. 1:** Comparison of  $\bar{q}_i$  and  $Q(x)$  for  $\beta = 10$



**Fig. 2:** Proof of Lemma 1

*Proof:* We begin by noting that both  $a$  and  $A$  are decreasing functions, because  $f$  is negative. Reasoning by contradiction, suppose that  $A(i) \geq a_i$  but  $A(i+1) < a_{i+1}$ . Therefore, there exists an  $x \in (i, i+1)$  such that  $A(x)$  intersects the straight line joining points  $(i, a_i)$  and  $(i+1, a_{i+1})$ , as illustrated in Figure 2. The slope of this line at  $x$  is  $f(a_i)$  and the slope of  $A$  at point  $x$  is  $f(A(x))$ . At the intersection we must have  $f(a_i) > f(A(x))$ . But  $a_i > A(x)$  implies  $f(a_i) < f(A(x))$ , a contradiction.  $\square$

**Corollary 1**

$$\bar{q}_i \leq Q(i) \quad \forall i \geq 1. \quad (20)$$

Using this, we can rewrite equation (11) to obtain the following upper bound for the variance:

$$\sigma_\alpha^2 \leq \frac{2}{\alpha} \sum_{i \geq 1} Q(i) - \mu_\alpha - \mu_\alpha^2 \quad (21)$$

To approximate the summation, we use Euler's summation formula [10],

$$\sum_{i \geq 1} Q(i) = \int_1^\infty Q(x) dx + \sum_{k=1}^m \frac{B_k}{k!} (Q^{(k-1)}(\infty) - Q^{(k-1)}(1)) + R_m, \quad (22)$$

where the  $B_k$  are the Bernoulli numbers ( $B_0 = 1, B_1 = -\frac{1}{2}, B_2 = \frac{1}{6}, B_3 = 0, B_4 = -\frac{1}{30}, \dots$ ). From [10] Exercise 1.2.11.2-3, we know that for even  $m$ , if  $Q^{(m)}(x) \geq 0$  for  $x \geq 1$  then

$$|R_m| \leq \left| \frac{B_m}{m!} (Q^{(m-1)}(\infty) - Q^{(m-1)}(1)) \right|. \quad (23)$$

We note that, as  $x \rightarrow \infty$ , all derivatives of  $Q(x)$  tend to zero, because they all contain the factor  $f(Q(x))$ , by repeated differentiation of equation (17), and since  $Q(\infty) = 0$ , we have  $f(Q(\infty)) = f(0) = 0$ .

In our case, we will apply this formula with  $m = 2$ . We note that  $Q(1) = \bar{q}_1 = \alpha\mu_\alpha$  and  $Q'(1) = f(Q(1)) = f(\bar{q}_1) = \Delta\bar{q}_1 = -\bar{q}_1 = -\alpha$ . Furthermore,  $Q^{(2)}(x) \geq 0$  for  $x \geq 1$  because  $Q'(x) = f(Q(x))$  is an increasing function. Therefore, we have

$$\sum_{i \geq 1} Q(i) = \int_1^\infty Q(x) dx + \frac{1}{2}Q(1) - \frac{1}{12}Q'(1) + R_2 \leq \int_1^\infty Q(x) dx + \frac{1}{2}\alpha\mu_\alpha + \frac{1}{6}\alpha \quad (24)$$

and therefore the bound for the variance can be written as

$$\sigma_\alpha^2 \leq \frac{2}{\alpha} \int_1^\infty Q(x) dx + \frac{1}{3} - \mu_\alpha^2 \quad (25)$$

Note that, until now, we have not made use of the specific form of the function  $Q(x)$ . Using now formulas (18) and (12), we obtain the following upper bound for the variance:

**Theorem 2** *Under the asymptotic model for an infinite  $\alpha$ -full hash table with random probing and RH collision resolution discipline, the variance of the search cost of a random element satisfies (with  $\beta = 1/(1-\alpha)$ )*

$$\sigma_\alpha^2 \leq \frac{\pi^2}{3} + \frac{1}{3} + O\left(\frac{\ln \beta}{\beta}\right). \quad (26)$$

□

This gives us an upper bound of  $3.6232\dots$  for the variance of Robin Hood Hashing. Although a numerically computed value of approximately 1.883 has been known for a long time, this is the first proof that this variance is bounded by a small constant as  $\alpha \rightarrow 1$ . As Celis *et al.* observed, the fact that the variance is very small can be used to carry out a more efficient *mean-centered search*. If we call  $X$  the random variable “search cost of a random key” the expected cost of this modified search is  $\Theta(\mathbb{E}|X - \mu_\alpha|)$ . But Jensen’s inequality implies that

$$\mathbb{E}|X - \mu_\alpha| = \mathbb{E}\sqrt{(X - \mu_\alpha)^2} \leq \sqrt{\mathbb{E}(X - \mu_\alpha)^2} = \sigma_\alpha \quad (27)$$

so, the *mean value* of the search cost of a mean-centered search is proportional to the *standard deviation* of the cost of a standard search. Theorem 2 then implies that this search algorithm runs in expected constant time in a full table.

### 3.2 Bounding the tail of RH

We focus now on the tail of the distribution of the search cost, i.e. we study

$$\Pr\{X \geq i\} = \bar{p}_i = \frac{1}{\alpha} \bar{q}_i = \frac{\beta}{\beta - 1} \bar{q}_i. \quad (28)$$

We proved earlier that  $\bar{q}_i \leq Q(i)$ . By applying  $f$  to both sides and recalling that  $f$  is a decreasing function, we have  $f(\bar{q}_i) \geq f(Q(i))$ . Using equations (16) and (17), we have  $\Delta \bar{q}_i = -\bar{q}_i \geq Q'(i)$ , and therefore

$$\Pr\{X \geq i\} \leq -\frac{\beta}{\beta - 1} Q'(i) = \frac{\beta}{\beta - 1 + e^{i-1}}. \quad (29)$$

If we take the upper bound as the tail  $\frac{\beta}{\beta - 1 + e^{x-1}}$  of a continuous probability function, its density function would be

$$p(x) = \frac{\beta e^{x-1}}{(\beta - 1 + e^{x-1})^2}, \quad (30)$$

which is symmetric around its mean (and mode) located at the point  $x$  such that  $e^{x-1} = \beta - 1$ , i.e.,  $x = 1 + \ln(\beta - 1)$ .

As a consequence, by equation (29), the probability that the search cost will exceed this amount by a given number of steps  $k$ :

$$\Pr\{X \geq 1 + \ln(\beta - 1) + k\} \leq \frac{\beta}{\beta - 1} \frac{1}{e^k + 1} \rightarrow \frac{1}{e^k + 1} \quad (31)$$

as  $\beta \rightarrow \infty$ .

Therefore, as the table becomes full, the mean moves to the right without bound, but the distribution remains tightly packed to the right of the mean, and the probability that the search cost exceeds the mean by a given amount decreases exponentially with the distance.

Finally, it is interesting to note that if we shift to the left the density function (30) so it is centered around zero, we obtain

$$p(1 + \ln(\beta - 1) + x) = \frac{\beta}{\beta - 1} \frac{e^x}{(1 + e^x)^2} \quad (32)$$

which, as  $\beta \rightarrow \infty$ , converges to  $\frac{e^x}{(1 + e^x)^2}$ , or, equivalently,  $\frac{e^{-x}}{(1 + e^{-x})^2}$ , the density function of a Logistic(0,1) distribution.



## 4 Analysis with deletions

We assume a process where we first insert keys until the table reaches load factor  $\alpha$ , and then we enter an infinite cycle where we alternate one random insertion followed by one random deletion.

If the distribution of the retrieval cost is given by  $p_i(\alpha)$  and a random element is inserted, the effect is described by equation (5). If we then perform a random deletion, the following classical lemma[6] shows that the distribution remains unchanged:

**Lemma 2** *Suppose a set contains  $n$  balls of colors  $1, 2, \dots, k$ , such that the probability that a ball chosen at random is of color  $i$  is  $p_i$ . Then, if one ball is chosen at random and discarded, the a posteriori probability that a random ball is of color  $i$  is still  $p_i$ .*

*Proof:* Call  $p'_i$  the probability that a random ball is of color  $i$  after the deletion. The expected number of balls of color  $i$  afterwards is  $(n-1)p'_i$ , but that number can also be obtained as the expected number before,  $np_i$ , minus the expected number of balls of color  $i$  lost, i.e.,

$$(n-1)p'_i = np_i - 1 \cdot p_i. \quad (33)$$

The result follows.  $\square$

Therefore, equation (5) describes also the probability distribution after one insert-delete step. Now, assume the process reaches a steady state. In that case, the distribution after the insert-delete must be equal to the distribution before, i.e.  $p_i(\alpha + \frac{1}{m}) = p_i(\alpha)$ , and replacing this in (5) we have

$$p_i(\alpha) = t_i(\alpha) - t_{i+1}(\alpha). \quad (34)$$

and equivalently,

$$\bar{p}_i(\alpha) = t_i(\alpha). \quad (35)$$

These equations play the role that equation (7) did for the case without deletions. Taking tails in both sides of this equation and setting  $i = 1$ , we can obtain the expected search cost  $\mu_\alpha$  as

$$\mu_\alpha = \bar{p}_1 = \bar{t}_1 = \frac{1}{1-\alpha}, \quad (36)$$

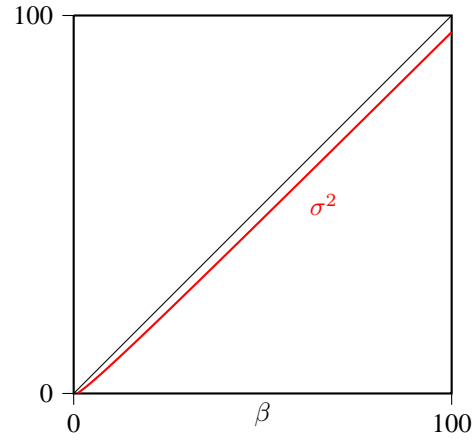
confirming the prediction that the expected successful search cost should approach the expected *unsuccessful* search cost when deletions are allowed.

For RH, from (35) we get  $\bar{p}_i = \bar{t}_i$ , and combining this with (13) we obtain

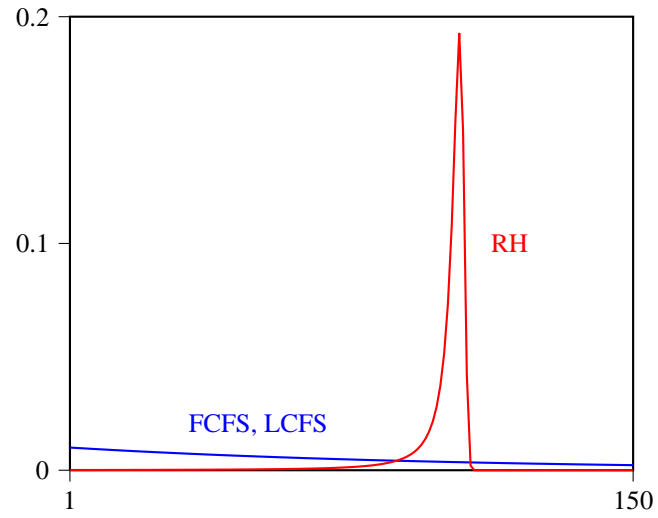
$$\bar{p}_1 = \frac{1}{1-\alpha}, \quad \bar{p}_{i+1} = \frac{\alpha \bar{p}_i^2}{1 + \alpha \bar{p}_i} \quad (37)$$

We can use this recurrence to compute numerically the distribution for RH.

Figure 3 shows the value of the variance of RH as a function of  $\beta = 1/(1-\alpha)$ , and from the plot we may see that the variance is very close to  $\beta$ . Moreover, Figure 4 shows the distribution of the search cost for the three methods, for  $\alpha = 0.99$ . As proven in [13] it can be seen that FCFS and LCFS are now identical and have very large dispersion ( $\sigma_\alpha^2 = \frac{\alpha}{(1-\alpha)^2}$ ), while RH retains a much more concentrated shape. We prove that this is indeed the case.



**Fig. 3:** The variance of RH with deletions as a function of  $\beta$



**Fig. 4:** Distribution of search costs for FCFS, LCFS and RH for  $\alpha = 0.99$

#### 4.1 Bounding the variance of RH with deletions

We begin by rewriting the recurrence equation (37) as

$$\bar{q}_1 = \beta - 1, \quad \Delta \bar{q}_i = -\frac{\bar{q}_i}{1 + \bar{q}_i} \quad (38)$$

This equation is of the form  $\Delta \bar{q}_i = f(\bar{q}_i)$  for  $f(x) = -\frac{x}{1+x}$ , and all the conditions required in section 3.1 are satisfied, so we can apply the exact same technique used there. Solving the associated differential equation

$$Q'(x) = f(Q(x)), \quad Q(1) = \beta - 1 \quad (39)$$

we find the solution

$$Q(x) = W((\beta - 1)e^{\beta - x}), \quad (40)$$

where  $W$  is Lambert's function satisfying  $x = W(x)e^{W(x)}$ . As a consequence, proceeding as in the proof of Theorem 2, we obtain the following result:

**Theorem 3** *Under the asymptotic model for an infinite  $\alpha$ -full hash table with random probing and RH collision resolution discipline, in the steady state of a sequence of insert-delete operations, the variance of the search cost of a random element satisfies (with  $\beta = 1/(1 - \alpha)$ )*

$$\sigma_\alpha^2 \leq \beta + \frac{1}{3} = \frac{1}{1 - \alpha} + \frac{1}{3}. \quad (41)$$

□

This proves our earlier conjecture that the variance was very close to  $\frac{1}{1 - \alpha}$ .

## 5 Acknowledgements

We are grateful to the anonymous reviewers, for their valuable comments and suggestions, that helped us improve the paper.

## References

- [1] O. Amble and D. E. Knuth. Ordered-hash-tables. *Computer Journal*, 17(2):135–142, 1974.
- [2] R. P. Brent. Reducing-the-retrieval-time-of-scatter-storage-techniques. *CACM*, 16(2):105–109, 1973.
- [3] P. Celis. *Robin Hood Hashing*. PhD thesis, University of Waterloo, 1986. Technical Report CS-86-14.
- [4] P. Celis, P.-Å. Larson, and J.I. Munro. Robin Hood Hashing. In *26th IEEE Symposium on the Foundations of Computer Science*, pages 281–288, 1985.
- [5] Walter Cunto and Patricio V Poblete. Two hybrid methods for collision resolution in open addressing hashing. In *SWAT 88*, pages 113–119. Springer, 1988.

- [6] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, January 1968.
- [7] G. H. Gonnet and J. I. Munro. Efficient-ordering-of-hash-tables. *SIAM Journal on Computing*, 8(3):463–478, 1979.
- [8] Leo J. Guibas. The analysis of hashing techniques that exhibit k-ary clustering. *J. ACM*, 25(4):544–555, October 1978.
- [9] L.J. Guibas. *The analysis of hashing algorithms*. STAN-CS. Stanford University, 1976.
- [10] D.E. Knuth. *Art of Computer Programming Volume 1: Fundamental Algorithms*. Addison-Wesley Publishing Company, 1972.
- [11] D.E. Knuth. *The Art of Computer Programming vol. 3 Sorting and Searching*. Addison-Wesley Publishing Company, 1998.
- [12] M. Mitzenmacher. A new approach to analyzing robin hood hashing. Preliminary version in <http://arxiv.org/abs/1401.7616>, 2014.
- [13] P. Poblete and A. Viola. The effect of deletions on different insertion disciplines for hash tables. In *Brazilian Symposium on Graphs, Algorithms and Combinatorics (GRACO)*, 2001.
- [14] P.V. Poblete and J.I. Munro. Last-Come-First-Served Hashing. *Journal of Algorithms*, 10:228–248, 1989.